

REMARKS

In the Office Action dated August 27, 2002, the Examiner rejected claims 1, 10-11, and 17 under 35 U.S.C. 103(a) as being unpatentable over Chorn (U.S. Patent No. 6,275,843) in view of Abe et al. (U.S. Patent No. 6,052,695), and in further view of Simor (U.S. Patent No. 5,060,150) (Office Action [hereinafter "OA"], paragraph 1); rejected claims 2-7 and 18 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al. and Simor, and in further view of Hoffpauir (hereinafter "Hoffpauir 1") (U.S. Stat. Inv. Reg. No. H1,896) (OA, paragraph 2); rejected claims 8-9, 12-15, and 19-21 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al. and Simor, and in further view of Hoffpauir (hereinafter "Hoffpauir 2") (U.S. Stat. Inv. Reg. No. H1,918) (OA, paragraph 3); and rejected claim 16 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al., Simor, and Hoffpauir 2, and in further view of Jones et al. (U.S. Patent No. 5,193,110) (OA, paragraph 4). The Examiner also objected to the disclosure because of certain alleged informalities.

In view of the foregoing amendment and the remarks that follow, Applicant respectfully traverses the Examiner's objections to the disclosure and rejections of the claims under 35 U.S.C. § 103(a).

The Examiner objected to the disclosure as containing a non-descriptive title. Applicant has amended the title to the one suggested by the Examiner.

The Examiner also objected to the disclosure as containing a grammatical error on page 3. Applicant has amended the specification to correct the error.

The Examiner rejected claims 1, 10-11, and 17 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al., and in further view of Simor (OA, paragraph 1). The Examiner alleged that Chorn discloses at least a resource manager operable to control the allocation of a resource to competing computing processes, the resource manager being responsive to identification of a thread for a first process requesting resource allocation. The Examiner admitted that Chorn does not disclose a joining function that notifies a resource manager on termination of a thread for a second process but alleged that the teachings of Abe et al. suggest including a joining function. The Examiner also admitted that Chorn fails to disclose "that after termination of [the] thread for a second process, the resource of the first process is allocated." The Examiner relied on Simor to teach that feature.

Claim 1 provides for a resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

Applicant respectfully submits that Chorn in view of Abe et al., and in further view of Simor do not disclose or suggest at least this claimed combination of elements. For example, the references do not disclose or suggest at least a resource manager being

operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process.

Chorn discloses a system in which multiple service requests of a global transaction are processed by a single server application program instance without the use of multiple intermediate communication server instances (abstract). A client application program provides access to one or more resources that are required by a transaction (col. 6, line 64 - col. 7, line 1). A resource manager provides access to a resource for the application program (col. 7, lines 8-9). A communication resource manager controls communication between the client application program and other application programs that are participating in transactions (col. 7, lines 22-25). Thread identifiers are used by the communication resource manager to manage its resources for processes using the services of the communication resource manager (col. 14, lines 38-40).

In contrast, systems and methods consistent with the present invention with the present invention as recited for example in claims 1, 10-12, and 17, include a resource manager that is operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process, wherein the first process and the second process are call processing applications. The resource managers of Chorn do not allocate a resource to a thread for a first process responsive to termination of a thread for a second process. Accordingly, Chorn does not disclose, teach, or suggest a resource manager being operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process. Applicant also notes that the

Examiner admitted that Chorn does not allocate a resource after termination of a thread for a second process.

Abe et al. are not sufficient to overcome the deficiencies of Chorn. Abe et al. disclose a system in which a first log processing thread outputs all log data including that for a second log processing thread and returns an output completion notification, when the second log processing thread goes into a waiting state for the output completion notification (col. 27, lines 35-45). When the notification is received, the second log processing thread terminates its own processing (col. 27, lines 45-47). Abe et al. are silent on allocating a resource to a thread for a first process responsive to termination of a thread for a second process. Accordingly, Abe et al., either alone or in combination with Chorn, do not disclose, teach, or suggest a resource manager being operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process.

Simor is not sufficient to overcome the deficiencies of Chorn and Abe et al. Simor discloses a system in which a process termination monitor may be requested at each resource allocation for a process (col. 17, lines 55-57). Resource managers are accordingly notified when the process terminates, and the resource release can be initiated by the resource manager itself (col. 17, lines 57-59). This configuration disclosed by Simor is different than the presently claimed invention. For example, Simor does not allocate a resource to a thread for a first process responsive to the termination of a thread for a second process. At most, Simor can detect the termination of a process and release a corresponding resource. But this detection and release does not suggest that there is also an allocation of the resource to a different process

after the release. Accordingly, Simor, either alone or in combination with Chorn and/or Abe et al., does not disclose, teach, or suggest a resource manager being operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process.

For at least the foregoing reasons, Applicant submits that claim 1 is patentable over Chorn in view of Abe et al., and in further view of Simor. Applicant also submits that the teachings of Hoffpauir 1 are insufficient to remedy the deficiencies of Chorn, Abe et al., and Simor. Accordingly, the combination of Chorn, Abe et al., Simor, and Hoffpauir 1 does not teach or suggest claim 1. Because claims 10-11, and 17 are independent claims with limitations similar to those of claim 1, Applicant further submits that claims 10-11, and 17 are patentable over Chorn in view of Abe et al., and in further view of Simor for at least the reasons given with respect to claim 1.

The Examiner rejected claims 8-9, 12-15, and 19-21 under 35 U.S.C. 103(a) as being unpatentable over Chorn in view of Abe et al. and Simor, and in further view of Hoffpauir 2. Claim 12 provides for a telecommunications apparatus comprising at least one telephony resource for connection to a telecommunications network and a resource manager for controlling allocation of the telephony resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for the first process requesting allocation of the resource, when the resource is already allocated to a thread for the second process, to establish a joining function to the thread for the second process, the joining function being operable to notify the resource manager on termination of the thread for the second

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

Applicant respectfully submits that Chorn in view of Abe et al. and Simor, and in further view of Hoffpaur 2 do not disclose or suggest at least this claimed combination of elements. For example, the references do not disclose or suggest at least a resource manager being operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process.

As explained above with reference to claim 1, Chorn in view of Abe et al. and Simor do not teach a resource manager being operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process. Hoffpaur 2 is not sufficient to overcome the deficiencies of Chorn, Abe et al., and Simor. Hoffpaur 2 disclose a system in which an integrated authentication center includes an application process, such as a call processing application, that includes a plurality of software objects such as a home location register and an authentication center (abstract). The system includes a resource manager application that manages and allocates the resources of a resource assembly with respect to a call processor and enables different applications of the call processor to interface with resources of the resource assembly (col. 13, lines 62-66).

Hoffpaur 2 fails to show allocating a resource to a thread for a first process responsive to termination of a thread for a second process. Accordingly, Hoffpaur 2, either alone or in combination with Chorn, Abe et al., and Simor, do not disclose, teach, or suggest a resource manager being operable in response to termination of a thread for a second process to allocate a resource to a thread for a first process.

For at least the foregoing reasons, Applicant submits that claim 12 is patentable over Chorn in view of Abe et al. and Simor, and in further view of Hoffpaur 2. Applicant also submits that the teachings of Jones et al. are insufficient to remedy the deficiencies of Chorn, Abe et al., Simor, and Hoffpaur 2. Accordingly, the combination of Chorn, Abe et al., Simor, Hoffpaur 2, and Jones et al. does not teach or suggest claim 12.

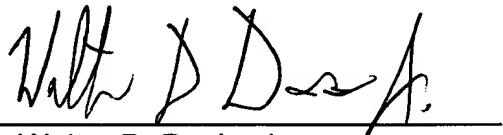
The dependent claims 2-9, 13-16, and 18-21 are allowable not only for the reasons stated above with regard to their respective allowable base claims, but also for their own patentable features.

Since each of the claims is allowable, Applicant respectfully requests the timely allowance of this application.

If an extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Amendment, such extension is requested. If there are any other fees due under 37 C.F.R. §§ 1.16 or 1.17 which are not enclosed herewith, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge such fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

By: 

Walter D. Davis, Jr.
Reg. No. 45,137

Dated: November 27, 2002

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

APPENDIX TO AMENDMENT OF NOVEMBER 27, 2002

AMENDMENTS TO THE SPECIFICATION:

Page 3, the paragraph beginning at line 16:

An embodiment of the invention is able to provide reliable resource management through the use of a joining function which enables the resource manager simply to join to the thread which is the current resource owner. Such a joining function is a standard language component of, for example, the Java™ language. Accordingly, a resource manager in accordance with an embodiment of the invention, can be provided with a single method "acquireDevice()" which may be called by any process, or application, wishing to use a resource (e.g., a telephony device such as a modem). Standard language primitives ensure that only one application may execute this method at one time, thereby guaranteeing exclusive access. The would-be owning application can identify its operating thread to the resource manager. By [effecting] affecting the join on the thread which is the current resource owner, the resource manager is able simply to wait until the owning application thread terminates, at which time the join language function notifies the resource manager that the previously owning application has terminated. This notification is provided by the language constructs irrespective of the manner in which the previous owning application terminated. The resource manager is then able to allocate the resource to the requesting application.

FINNEGAN
ENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

AMENDMENTS TO THE CLAIMS:

1. (Amended) A resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for [a]the first process requesting allocation of the resource, when the resource is already allocated to a thread for [a]the second process, to establish a joining function to the thread for the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

3. (Amended) The resource manager of claim 2, wherein the [processes] first and second processes are software applications operable in the object oriented environment.

7. (Amended) The resource manager of claim 1, [where]wherein the join function is a join of the type provided in a Java language environment, [wherein]and a language event passively releases a resource on termination of a thread identified by the join function.

9. (Amended) The resource manager of claim 8, comprising a dispatch mechanism for controlling dispatching of a call received by the telephony device to the telecommunications applications.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

10. (Amended) A resource manager operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager comprising means responsive to identification of a thread for [a]the first process requesting allocation of the resource, when the resource is already allocated to a thread for [a]the second process, to establish a joining function to the thread for the second process and means responsive to the joining function notifying the resource manager on termination of the thread for the second process to allocate the resource to the thread for the first process.

11. (Amended) A computer software resource manager on a data carrier, the resource manager being operable to control allocation of a resource to competing computing processes including at least a first process and a second process, the resource manager being responsive to identification of a thread for [a]the first process requesting allocation of the resource, when the resource is already allocated to a thread for [a]the second process, to establish a joining function to the thread for the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

12. (Amended) Telecommunications apparatus comprising at least one telephony resource for connection to a telecommunications network and a resource manager for controlling allocation of the telephony resource to competing computing processes

including at least a first process and a second process, the resource manager being responsive to identification of a thread for [a]the first process requesting allocation of the resource, when the resource is already allocated to a thread for [a]the second process, to establish a joining function to the thread for the second process, the joining function being operable to notify the resource manager on termination of the thread for the second process, and the resource manager being operable in response to termination of the thread for the second process to allocate the resource to the thread for the first process.

16. (Amended) The telecommunications apparatus of claim [15]12, wherein the call processing applications comprise at least one application selected from:

- a call answering application;
- a voicemail application;
- a facsimile application; and
- a data application.

17. (Amended) A computer-implemented method of managing allocation of a resource to competing processes including at least a first process and a second process, the method including:

responding to identification of a thread for [a]the first process requesting allocation of the resource, when the resource is already allocated to a thread for [a]the second process, to establish a joining function to the thread for the second process;

responding to the joining function notifying termination of the thread for the second process to allocate the resource to the thread for the first process.

18. (Amended) The method of claim 17, wherein the join function is a join function of the type provided in a Java language environment, [whereby]and a language event passively releases a resource on termination of a thread identified by the join function.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com